KATHOLIEKE UNIVERSITEIT
LEUVEN

Faculty of Business and Economics

Project portfolio management: capacity allocation, downsizing decisions and sequencing rules

J. Herbots, W. Herroelen and R. Leus

DEPARTMENT OF DECISION SCIENCES AND INFORMATION MANAGEMENT (KBI)

KBI 0826

Project portfolio management: capacity allocation, downsizing decisions and sequencing rules

Jade Herbots
Department of Decision Sciences and Information Management
Katholieke Universiteit Leuven, Belgium
Tel. +32 16 32 67 57; Fax +32 16 32 66 24
Jade.Herbots@econ.kuleuven.be

Willy Herroelen
Department of Decision Sciences and Information Management
Katholieke Universiteit Leuven, Belgium
Tel. +32 16 32 69 70; Fax +32 16 32 66 24
Willy.Herroelen@econ.kuleuven.be

Roel Leus
Department of Decision Sciences and Information Management
Katholieke Universiteit Leuven, Belgium
Tel. +32 16 32 69 67; Fax +32 16 32 66 24
Roel.Leus@econ.kuleuven.be

— November 2008 —

Project portfolio management: capacity allocation, downsizing decisions and sequencing rules

This paper aims to gain insight into capacity allocation, downsizing decisions and sequencing rules when managing a portfolio of projects. By *downsizing*, we mean reducing the scale or size of a project and thereby changing the project's content. In previous work, we have determined the amount of critical capacity that is optimally allocated to concurrently executed projects with deterministic or stochastic workloads when the impact of downsizing is known. In this paper, we extend this view with the possibility of *sequential* processing, which implies that a complete order is imposed on the projects. When projects are sequenced instead of executed in parallel, two effects come into play: firstly, unused capacity can be shifted to later projects in the same period; and secondly, reinvestment revenues gain importance because of the differences in realization time of the sequenced projects. When project workloads are known, only the second effect counts; when project workloads are stochastic, however, the project's capacity usage is uncertain so that unused capacity can be shifted to later projects in the same period. In this case, both effects need to be taken into account. In this paper, we determine optimal sequencing rules when the selection and capacity-allocation decisions for a set of projects have already been made. We also consider a combination of parallel and sequential planning and we perform simulation experiments that confirm the appropriateness of our capacity-allocation methods.

Keywords: project portfolio management; downsizing; sequencing.

1. Introduction

This paper aims to gain insight into capacity allocation, downsizing decisions and sequencing rules when managing a portfolio of projects. By *downsizing*, we mean reducing the scale or size of a project and thereby changing the project's content. Downsizing is based on the principle that project selection is not an all-or-nothing decision, but that multiple funding alternatives exist for executing a project (Sharpe & Keelin 1998, Kavadias & Loch 2004). *Project portfolio management* deals with the continuous flow of projects; it entails choosing the right projects and the associated capacity allocation, as well as the prioritization of projects. The latter problem coincides with finding an appropriate *sequencing* rule, when projects are not concurrently executed. We refer to sequential processing when a complete order is imposed on the projects.

This paper considers environments where capacity expansion through the deployment of additional resources is not possible, such as, for instance, most R&D (Research and Development) or NPD (New Product Development) departments, where the number of researchers or other critical resources is fixed as a result of a strategic decision depending on the revenue stream and costs. The company's restricted resource availability then represents the most important constraint on project selection.

Maximizing the overall value of the selected projects is our primary objective. We assume that the revenue obtained from a single project depends on its strategic and financial value, its allocated capacity and realization time (in case of reinvestment revenues). The main project characteristics are the yield function, the project workload or duration and *downsizeability*. The project workload is estimated in terms of the number of manhours of capacity required during every period of the problem horizon and may be either deterministic or stochastic. In the case of sequential project execution, the workload is a measure for the project duration. When a project's workload or duration is deterministic, the allocated capacity fixes the scale of the project before its start, while in case of stochastic workloads, the project is only downscaled if the actual workload exceeds the allotted capacity. The downsizeability expresses the effect of such a scale reduction on the revenue obtained from a project.

In previous work (Herbots et al. 2008), we have determined the amount of critical capacity that is optimally allocated to concurrently executed projects with deterministic or stochastic workloads when the impact of downsizing is known. In this paper, we extend this view with the possibility of sequential processing. When projects are sequenced instead of executed in parallel, two effects come into play: firstly, unused capacity can be shifted to later projects in the same period; and secondly, reinvestment revenues gain importance because of the differences in realization time of the sequenced projects. When project workloads are known, only the second effect counts; when project workloads are stochastic, however, the project's capacity usage is uncertain so that unused capacity can be shifted to later projects in the same period. In this case, both effects need to be taken into account.

Although the revenue of a project may vary with its realization time (Kerzner 1997, Chen & Askin 2009), project-specific benefits from early completion are not taken into account in this paper and we apply the risk-free rate (see Brealey & Myers (2003)) as the discount rate to calculate reinvestment revenues for all projects. In multi-project environments, typically a combination of projects executed in parallel and in sequence will occur (see, for instance, Wheelwright & Clark (1992), Adler et al. (1995)). A discussion on the combination of both planning methods is included at the end of this paper.

The scope of this paper is twofold. Firstly, we verify the validity of the selection and capacity-allocation methods for parallel projects developed in Herbots et al. (2008) and summarized briefly in Section 2. More specifically, we test whether the obtained methods for allocating capacity to concurrent projects remain suitable when projects are executed sequentially through a number of simulation experiments. We consider both projects with deterministic and stochastic workloads and cases with and without reinvestment revenues; an elaborate overview of our experiments is given in Section 3. Section 4 contains some references to the literature on the topic of project selection and sequencing.

The second objective of this paper is to determine appropriate sequencing rules when the selection and capacity-allocation decisions for a set of projects have already been made; sequencing in case of reinvestment revenues is the topic of Section 5. The setup and results of our computational experiments are laid out in Section 6. A discussion on the combination of parallel and sequential execution is included in Section 7; we end the paper with some conclusions in Section 8.

2. Project selection, capacity-allocation and downsizing decisions

The selection and allocation process has been extensively described in Herbots et al. (2008); it consists of three phases that will briefly be discussed in this section. During the first phase, discussed in Section 2.1, a company identifies all project opportunities and their yield functions. Next, the resources are split between the selected projects (Section 2.2). In Phase III, we downsize running projects that exceed their allocated resources. The downscaling process is the topic of Section 2.3.

2.1 Phase I: Identify project opportunities and yield functions

During the first phase all project opportunities and their yield functions are identified. This yield function provides a link between a project's main input parameters, namely it's workload, downsizeability and allocated capacity and the project's value. A project's workload is expressed as the number of manhours the project requires during the problem horizon. The workload of a project k may be either deterministic or stochastic; in the stochastic case, the workload P_k is a random variable, with mean μ_k. We assume that P_k follows a uniform distribution with bound values equal to $\mu_k(1 \pm \beta_k)$, with $0 < \beta_k < 1$; these values can also be interpreted as confidence limits for the estimated value μ_k. When the workload is considered to be deterministic, P_k is replaced by μ_k. The resource budget available for projects is expressed as the available number M of manhours within each period of the planning horizon. Value $\hat{M}_k$ ($\hat{M}_k \leq M$) indicates the amount of capacity allotted to project k. We define the scaled input I_k as the apportioned project's workload μ_k, scaled so that the actually needed workload during execution fits the available capacity. The scaled input is a virtual measure that is used to model the financial impact of downsizing. If the realized project's workload P_k is smaller than or equal to the reserved capacity, the project can be executed at its full scale and the scaled input I_k is equal to μ_k. In case P_k exceeds $\hat{M}_k$, we need to downsize the project to a percentage $\hat{M}_k/P_k$ so that the scaled input $I_k = \mu_k \cdot \hat{M}_k/P_k$.

The yield function of project k is modeled as follows.

$$y_k(\hat{M}_k; \alpha_k, \gamma_k, \mu_k, C_k) = \alpha_k {I_k}^{\gamma_k} C_k, \tag{1}$$

with

$$\begin{cases} I_k = \mu_k & \text{if } P_k \leq \hat{M}_k, \\ I_k = \mu_k \hat{M}_k/P_k & \text{if } P_k > \hat{M}_k, \end{cases} \tag{2}$$

with $\alpha_k > 0$ the project's production factor (cfr. the Cobb-Douglas functional form of production functions (Cobb & Douglas 1928)), expressed in terms of profit per workload and C_k a constant value. If we set $C_k = {\mu_k}^{1-\gamma_k}$, then the profit per workload $y_k(\hat{M}_k; \alpha_k, \gamma_k, \mu_k, C_k)/\mu_k$ of the project when it is not downsized is equal to $\frac{\alpha_k {\mu_k}^{\gamma_k} {\mu_k}^{1-\gamma_k}}{\mu_k}$, which corresponds to the production factor α_k. The use of the Cobb-Douglas function is motivated by its flexibility; the marginal returns can be made increasing, linear or decreasing, depending on the value of γ_k.

The impact of downscaling on a project k's overall value depends on the downsizeability parameter γ_k. When $0 < \gamma_k < 1$, the impact of a resource reduction is moderate; which will often hold for *incremental innovation projects* (Tushman et al. 1997), which aim to improve existing products and result in moderate productivity increases. When $\gamma_k = 1$, a linear relation between in- and output is implied. In case $\gamma_k > 1$, the effect of downsizing is detrimental to the project's revenue. This behavior is inherent to many *radical innovation projects*, which lead to radical improvements or completely new products (Tushman et al. 1997).

We illustrate the effect of an overrun of the allocated capacity and the appropriate downscaling on a small example, visualized in Figure 1. We consider a project with a workload that varies around an average load of 5 ($\mu_k = 5, \beta_k = 0.2$) and an allocated capacity of 4.5. The project's production factor and the constant factor are both equal to 1; the downsizeability parameter is 2. If we monitor that the realized workload P_k will lead to an overrun

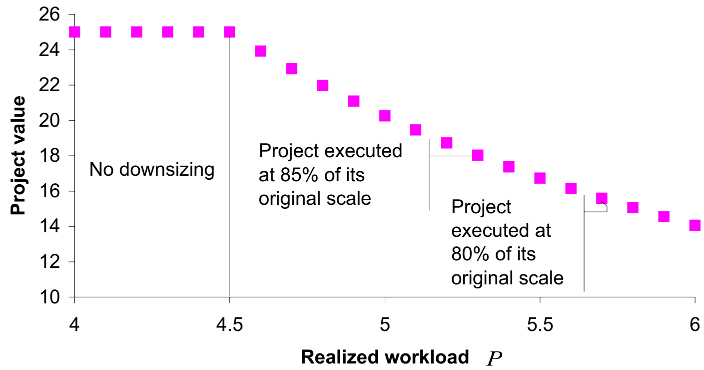

Figure 1: Project values for a project k with an average workload of 5 and an allocated capacity $\hat{M}_k$ of 4.5

of the allocated capacity, downsizing will need to take place. For instance, if $P_k = 5.3$, the project needs to be reduced with a scale factor equal to $\hat{M}_k/P_k = 4.5/5.3 = 85\%$. Instead of reaping 25, which is the revenue of the full-scaled project, the company will receive $(85\% \cdot 5)^2 = 18.06$ for the downsized project.

2.2 Phase II: Allocate resources

Next, we select the projects to be executed in the next period and determine the allocated capacity. During concurrent project execution, resources can only be employed by the projects they were initially assigned to, whereas in case of sequential execution, unused resources can be shifted to later projects. We assume that all revenues are reaped at the project realization time.

In Herbots et al. (2008) we have established the optimal selection by comparing the objective values of all feasible selections (through full enumeration). A selection is feasible if the sum of the lower bounds of the projects in the selection is smaller than or equal to the available capacity M in a period. Determining an optimal capacity allocation for a set of $\hat{N}$ projects corresponds to solving a non-linear program (NLP) of the following form:

$$\max z = \sum_{k=1,\dots,\hat{N}} E[y_k(\hat{M}_k)] \tag{3}$$

subject to

$$\begin{array}{lll} \sum_{k=1,\dots,\hat{N}} \hat{M}_k & \leq & M \\ \hat{M}_k & \leq & \overline{L}_k \quad (k = 1, \dots, \hat{N}) \\ \hat{M}_k & \geq & \underline{L}_k \quad (k = 1, \dots, \hat{N}). \end{array}$$

The allocated capacity $\hat{M}_k$ is bounded by a lower limit $\underline{L}_k$ and an upper limit $\overline{L}_k$. The presence of a lower downsizeability limit incorporates the fact that a project is not viable below a minimal resource input. The upper bound represents the resource input above which capacity increases cease to produce additional profits.

2.3 Phase III: The downscaling process

When the project workloads are deterministic, the project's scale is a direct consequence of the allocated capacity and is known before the project's start. In such cases, the flexibility is maximal and we will assume in this text that $\underline{L}_k = 0$ and $\overline{L}_k = \mu$. In the stochastic case, the actual required amount of resources (i.e. the realization of P_k) is only revealed during the execution of project k. If this exceeds the foreseen capacity, downsizing occurs. Since the need to downsize is only revealed during the execution of the project, the lower downsizeability bound $\underline{L}_k$ is set to $\mu_k(1-\beta_k)$; the upper bound $\overline{L}_k$ is equal to $\mu_k(1+\beta_k)$.

3. Problem statement

Four experiments are performed in this paper, one for each of the different considered environments (deterministic versus stochastic and reinvestment revenues versus no reinvestment revenues). In each of these tests or simulation experiments, we will compare two policies to assess the quality of the capacity-allocation methods developed in Herbots et al. (2008). A policy's quality is measured by the rewards reaped at the end of a period.

The first policy appoints an amount of capacity to project k, that results from the methods in Herbots et al. (2008), so that $\hat{M}_k \in [\underline{L}_k, \overline{L}_k]$, with $\underline{L}_k$ and $\overline{L}_k$ respectively the lower and upper bound on the allocated capacity of project k.

The second policy does not limit the capacity of an individual project and will only downsize the final projects in the sequence if necessary: $\hat{M}_k$ is equal to $\mu_k(1+\beta_k)$. When the workload is considered to be deterministic, it is represented by μ_k. In the stochastic case, the workload P_k is drawn uniformly from the interval $[\mu_k(1-\beta_k), \mu_k(1+\beta_k)]$ with mean μ_k.

We perform the following four experiments. First, we test whether the exact method summarized in Herbots et al. (2008) performs better than a common 'rule of thumb' for selection and resource allocation. In the case of deterministic workloads and no reinvestment revenues, the selection and capacity-allocation methods developed in Herbots et al. (2008) provide us with an optimal solution since the capacity usage of every project is constant. Moreover, no reinvestment revenues are reaped so that changing the sequence of project execution does not affect the total portfolio rewards. We compare the optimal portfolio with the one obtained by allocating resources in order of decreasing project production factor α. This approach corresponds to the popular productivity-index method (Cooper et al. 2001) and is a greedy heuristic for the traditional knapsack problem (Dantzig 1957). In addition, it represents an exact method for the fractional knapsack problem, which corresponds to our problem if all projects have linear marginal returns and zero lower bounds. Under this second policy, $\hat{M}_k$ may only be strictly smaller than μ_k for one selected project (with $\underline{L}_k = 0$), so that only the final project in the sequence may undergo a resource cut.

In the second experiment, reinvestment revenues are taken into account. We examine how the difference in portfolio value between the policies evolves when a sequencing rule is applied that maximizes the reinvestment revenues. Here, we use the same selection and capacity-allocation policies as in the first experiment. The presence of reinvestment revenues, however, calls for an appropriate sequencing rule. The derivation of the optimal sequencing rule is the subject of Section 5.

Our third experiment involves projects with unknown workloads and no reinvestment revenues. We test whether the methods for capacity allocation from Herbots et al. (2008) remain suitable when projects are executed sequentially. We again compare two policies: the first one selects projects and allocates resources as in Herbots et al. (2008). The second policy starts from the same selection, but allocates the amount of capacity $\overline{L}_k$ to every selected project k. Downsizing may occur on two occasions; namely, if the allocated capacity $\hat{M}_k$ is insufficient to allow for the full-scaled execution of project k (only under the first policy) or, on the other hand, if the periodic capacity is exhausted. In the latter case, the projects planned at the end of the planning period will be downsized. The applied sequencing rule in this environment ranks projects in order of decreasing downsizeability parameter γ_k to limit the financial impact of downsizing. If capacity is left over at the end of the period, we may add an additional project to the portfolio. This extra project is the one with the highest return on investment (when the project is not downsized) among the non-executed projects and is started if at least its minimum required workload is available. This rule is applied for both policies.

Finally, the fourth experiment compares the selection and capacity-allocation methods for the policies from the previous experiment in the presence of reinvestment revenues. The tested hypothesis is again whether the capacity-allocation methods developed by Herbots et al. (2008) perform well when projects are sequenced instead of executed in parallel. The applied sequencing rule is presented in Section 5.

4. Literature

Excellent literature surveys on the topic of order acceptance and project selection are provided by Rom & Slotnick (2008), Guerrero & Kern (1998), Keskinocak & Tayur (2004) and Roundy et al. (2005). In this section we survey the different perspectives that have been developed on the topic by different researchers, with a particular focus on the work in single-machine environments, since this setting corresponds to the sequencing problem discussed in this paper. We refer to Pinedo (2008) for an overview of sequencing rules for machine problems, both for the case of deterministic and stochastic job processing times as well as for a broad variety of objective functions. Further in this section, we discuss the research on simultaneous selection and sequencing. First, we consider methods that maximize rewards of selected projects (with deterministic project characteristics), combined with tardiness or lateness penalties. Subsequently, we consider alternative objective functions and extensions to random project characteristics.

Gupta et al. (1992) develop an efficient polynomial-time dynamic programming method that solves the project selection and sequencing problem (a fixed number of projects is selected from a candidate set) while maximizing the net present value of the total return. Slotnick and Morton study the single machine job-selection and sequencing problem with deterministic job processing times and job rewards; their objective is to maximize the rewards in case of lateness (Slotnick & Morton 1996) and tardiness (Slotnick & Morton 2007) penalties. A pseudo-polynomial time algorithm to solve the former problem was developed by Ghosh (1997). The problem was extended by Lewis & Slotnick (2002) to multiple periods for the case where rejecting a job will result in no future jobs from that customer. An exact approach for solving the single-period weighted tardiness problem was presented by Slotnick & Morton (2007). Since it can only deal with very moderate problem instances, suboptimal algorithms were also presented. Other heuristics include genetic algorithms (Rom & Slotnick 2008) and greedy algorithms for selection and ordering problems (Alidaee et al. 2001). Yang & Geunes

(2007) consider an extension of the problem where job processing times are reducible at a cost and every job has a release time. In their paper, Yang and Geunes develop an exact algorithm for maximizing schedule profit for a given sequence of jobs, along with heuristics to solve the entire problem.

Other objective functions for the selection and sequencing problem have been considered in literature. Engels et al. (2003) seek to minimize the sum of the weighted completion times of the scheduled jobs and the total rejection penalty of the rejected jobs. A related objective function is used by Lu et al. (2008) for the unbounded parallel batch machine scheduling problem with release dates. When a job is selected, it is added to the batch and otherwise a rejection penalty is incurred. The parallel batch machine can process a number of jobs simultaneously so that the makespan is the same for all jobs in a batch. The authors aim to minimize the total makespan in combination with the rejection penalties. A related problem is the job interval selection problem (JISP) where a job is determined by a set of intervals. Chuzhoy et al. (2001) consider some special cases of the JISP. The paper develops algorithms that aim to maximize the number of jobs scheduled between their release dates and deadlines.

The following articles include uncertainty in the project characteristics. Blau (1973) investigates the problem of sequencing n jobs on one machine, while looking at different types of tardiness; the processing times and due dates for the jobs are random. De et al. (1991) study the sequencing problem and minimize the weighted number of tardy jobs. De et al. (1991) consider the case where a project-dependent cost is charged when starting the execution of a project so that it becomes a selection problem. It is assumed that a revenue is reaped at the completion time of a project. The goal is to maximize the expected rewards of a selection of projects with random processing times and random deadlines.

5. Project sequencing in case of reinvestment revenues

In this section, we develop a sequencing rule for a set of projects that have been allocated an amount of capacity for the case where reinvestment revenues are reaped. The index that determines the best sequence is closely related to the *weighted discounted shortest expected processing time* (WDSEPT) rule (Pinedo 2008). An introduction to the WDSEPT rule is given in Section 5.1; the sequencing rule applied in our computational experiments is presented in Section 5.2.

5.1 The weighted discounted shortest expected processing time

For a set of projects or jobs k with weight w_k and duration $\hat{d}_k$, the WDSEPT rule minimizes the expected weighted sum of the discounted completion times $E[\sum w_k(1 - e^{-i\hat{d}_k})]$, in the class of non-preemptive static list policies[1] and non-preemptive dynamic policies[2] for the stochastic version of the one-machine job-sequencing problem $1||\sum w_k(1 - e^{-i\hat{d}_k})$ with arbitrary processing times (Pinedo 2008). The WDSEPT rule sequences jobs in decreasing order of the ratio

$$\frac{w_k E[e^{-i\hat{d}_k}]}{1 - E[e^{-i\hat{d}_k}]}, \tag{4}$$

with i the discount rate.

5.2 Sequencing rule

In this paper, we consider the case where all projects are executed in sequence. This implies that we can interpret variables and project characteristics such as $\hat{M}_k$, P_k and μ_k as time measures. For instance, for projects with deterministic workloads, the allocated capacity $\hat{M}_k$ can be used as a measure for the duration $\hat{d}_k$ of project k. In this case, the sequence that maximizes the weighted discounted expected profits of the projects can be found by sequencing the projects in decreasing order of the following rule, obtained from the WDSEPT rule.

$$\alpha_k \hat{M}_k^{\gamma_k} C_k \frac{e^{-i\hat{M}_k}}{1 - e^{-i\hat{M}_k}}, \tag{5}$$

with i the discount rate. The weight w_k in Eq. (4) corresponds to the project value $\alpha_k \hat{M}_k^{\gamma_k} C_k$.

Proposition 1. *The sequence that maximizes the weighted discounted expected profits of projects with stochastic workloads is determined by sequencing the projects in decreasing*

[1] defined by Pinedo (2008) as follows. 'Under a non-preemptive static list policy the decision maker orders the jobs at time zero according to a priority list. This priority list does not change during the evolution of the process and every time a machine is freed the next job on the list is selected for processing.'

[2] defined by Pinedo (2008) as follows. 'Under a non-preemptive dynamic policy, every time a machine is freed, the decision maker is allowed to determine which job goes next. His decision at such a point in time may depend on all the information available, e.g., the current time, the jobs waiting for processing, the jobs currently being processed on other machines and the amount of processing these jobs already have received on these machines. However, the decision maker is not allowed to preempt; once a job has begun its processing, it has to be completed without interruption.'

order of an index. The index of project k is

$$\frac{\int_{\mu_k(1-\beta_k)}^{\hat{M}_k} \frac{e^{-ip}\alpha_k C_k \mu_k^{\gamma_k-1}}{2\beta_k}\,dp + \int_{\hat{M}_k}^{\mu_k(1+\beta_k)} \frac{e^{-i\hat{M}_k}\alpha_k \mu_k^{\gamma_k-1}(\frac{\hat{M}_k}{p})^{\gamma_k} C_k}{2\beta_k}\,dp}{1-e^{-iE[\hat{d}_k(\hat{M}_k)]}}, \tag{6}$$

with $\hat{d}_k(\hat{M}_k)$ the duration of project k that has been assigned an amount of capacity $\hat{M}_k$.

The proof of Proposition 1 relies on an adjacent pairwise interchange argument and is given in the appendix.

The first part of the numerator in Eq. (6) corresponds to the discounted expected value of the non-downsized project, while the second part is the discounted expected value of the downsized project. The expected value of the completion time $\hat{d}_k(\hat{M}_k)$ of a project is determined as

$$E[\hat{d}_k(\hat{M}_k)] = \int_{\mu_k(1-\beta_k)}^{\hat{M}_k} \frac{p}{2\mu_k\beta_k}\,dp + \int_{\hat{M}_k}^{\mu_k(1+\beta_k)} \frac{\hat{M}_k}{2\mu_k\beta_k}\,dp. \tag{7}$$

The first part of Eq. (7) denotes the expected duration if the project is not downsized, while the second part represents the expected duration of the downsized project. When the allocated capacity is equal to a project's upper bound, its expected duration is equal to the average workload (expressed in time units) μ_k.

6. Computational experiments

In our computational experiments we generate sets of random projects for which the selection and sequence are computed according to different policies. We conduct four different experiments and evaluate the performance of the policies based on the project portfolio value. Section 6.1 provides a description of our experimental setup and the results are reported in Section 6.2.

6.1 Experimental setup

During each simulation experiment, we generate a set of random projects and determine the portfolio composition according to two policies. The performance of these policies is then evaluated through a comparison of the profit of the resulting portfolios. An overview of the factors varied during the simulation experiments is given in Table 1. The capacity M available for allocation is $5lN$, with $l \in [0, 1]$ the load parameter and $5N$ the joint average workload of the N projects. The number of periods for which we calculate the portfolio profits equals the number of replications of the simulation; we perform sufficient replications to ensure convergence of the results.

Table 1: Factors of the simulation experiment

factor	name	values
α	production factor	$U(0, 10)$
C	constant value	$\mu^{1-\gamma}$
μ	average workload	$U(0, 10)$
β	parameter of the workload distribution	$U(0, 1)$
N	number of available projects	20, 30
l	load parameter	0.6, 0.8
i	interest rate	0, 0.01

Our computational experiments are performed on a computer with a 1GHz Pentium III processor, the algorithms are coded in Microsoft Visual C++. The presented statistics issue from one-tailed paired tests, using simulation results from runs with the same seed; the significance level is 1% if not stated otherwise. In the following section, we report on the experimental results of the four experiments.

6.2 Experimental results

The computational results of the different experiments are reported in the subsequent paragraphs. In Section 6.2.1, we discuss the case of deterministic workloads and no reinvestment revenues; Section 6.2.2 deals with deterministic workloads and reinvestment revenues. Sections 6.2.3 and 6.2.4 cover the experiments on projects with stochastic workloads; reinvestment revenues are only considered in the latter section.

6.2.1 Deterministic workloads and no reinvestment revenues

As we have already mentioned in Section 3, the procedure laid out in Herbots et al. (2008) is optimal in case of deterministic workloads and no reinvestment revenues. In the first experiment, this procedure is compared to a second policy that selects projects in decreasing order of the project's return on investment (of the non-downsized project). We downsize the last project, if needed, to fit the period's available resources.

The results in Table 2 show the average portfolio values (i.e. the total reward obtained from all executed projects) for the optimal policy (P1) and the second policy (P2) and the average percentage increase when changing to the optimal policy for the test sets with the indicated factor values. These differences are all significant at the level of 1%. Clearly, a beneficial effect of the optimal policy is observed for all cases. For higher load values, the available capacity of the portfolio decreases, while the same set of projects remains available. Table 2 reports an increase in the average portfolio value for both policies. These results show that it pays off to apply more sophisticated policies to perform the project portfolio selection.

Table 2: Average portfolio values for the optimal policy (P1) and the second policy (P2) and average percentage increase when changing from P2 to P1 for deterministic workloads

N	l	Average portfolio value P1	Average portfolio value P2	Average percentage increase
20	0.6	478.05	367.04	30.25
	0.8	535.58	461.41	16.07
30	0.6	728.29	556.67	30.83
	0.8	825.4	707.84	16.61

6.2.2 Deterministic workloads and reinvestment revenues

In the second experiment, we consider projects with deterministic workloads and incorporate reinvestment revenues reaped during the time lag between the end of the project execution and the end of the period. The applied interest rate is equal to 0.01. The first policy determines the allocated amounts of capacity to the projects as in Section 6.2.1. The sequence is fixed based on the index from Eq. (5) and, as was explained in Section 3, downsizing may only occur at the end of the period.

The results in Table 3 show the average portfolio values for the first policy (P1) and the second policy (P2) and the average percentage increase when changing from P2 to P1 for deterministic workloads and reinvestment revenues. Results with an asterisk reveal no statistically significant (at the 1%-level) difference between the algorithms. Two asterisks reflect that no statistically significant result at the 5%-level could be found. The results reveal that the method for determining the capacity of parallel projects with known workloads performs better than the alternative policy for the case where projects are sequenced and reinvestment revenues are reaped. The difference between both policies is more explicit for higher load parameters. For these cases, the available capacity M has also been increased so that more projects are selected.

6.2.3 Stochastic workloads and no reinvestment revenues

In our third experiment we compare the value of the portfolio for two policies. Both policies determine a maximal amount of capacity that cannot be exceeded by the individual project's resource requirements. The first policy determines the project portfolio and the allocated capacity of projects through the procedures for parallel projects with uncertain workloads developed by Herbots et al. (2008), while the second policy selects the same projects but allocates capacity up to the projects' upper bounds. Projects are prioritized based on their downsizeability parameter γ; the details of both policies were previously laid out in Section 3.

The results in Table 4 indicate that restricting the allocated capacity as is done by the first policy, increases the pay-off even when projects are sequenced. The asterisk indicates that the difference between the algorithms was not statistically significant at the 1%-level, but only at the 5%-level. For high loads ($l = 0.8$), the available capacity is higher. Under these circumstances, more projects get selected and the differences between the policies become more significant.

Table 3: Average portfolio values for the first policy (P1) and the second policy (P2) and average percentage increase when changing from P2 to P1 for deterministic workloads and $i = 0.01$

N	l	Average portfolio value P1	Average portfolio value P2	Average percentage increase
20	0.6	556.31	515.28	7.96**
	0.8	837.74	746.78	12.18
30	0.6	988.76	947.79	4.32*
	0.8	1716.75	1495.11	14.82

Table 4: Average portfolio values for the first policy (P1) and the second policy (P2) and average percentage increase when changing from P2 to P1 for stochastic workloads

N	l	Average portfolio value P1	Average portfolio value P2	Average percentage increase
20	0.6	304.76	266.98	14.15*
	0.8	426.63	353.18	20.80
30	0.6	444.48	388.49	14.41
	0.8	633.34	507.57	24.78

6.2.4 Stochastic workloads and reinvestment revenues

The setup of this experiment and the applied policies only differ from the approach described in Section 6.2.3 in the way the projects are sequenced. Since reinvestment revenues are taken into account between the end of the project execution and the end of the period ($i = 0.01$), we apply the sequencing rule from Eq. (6). Table 4 shows that the average percentage increase of the portfolio value when changing from the second policy to the first policy for stochastic workloads and $i = 0.01$ is significantly (at the level of 1%) positive for all tested cases. This confirms the findings from Section 6.2.3 that restricting the allocated capacity of some projects leads to an increase in the pay-off, also when reinvestment revenues are taken into account.

Table 5: Average portfolio values for the first policy (P1) and the second policy (P2) and average percentage increase when changing from P2 to P1 for stochastic workloads and $i = 0.01$

N	l	Average portfolio value P1	Average portfolio value P2	Average percentage increase
20	0.6	406.16	366.02	10.96
	0.8	643.74	567.56	13.42
30	0.6	667.39	598.18	11.57
	0.8	1210.61	1052.8	14.99

7. Parallel and sequential planning combined

Although this paper focusses on the portfolio selection and prioritization of sequential projects, in multi-project environments, some projects may be planned in sequence while others are executed concurrently (cfr. Wheelwright & Clark (1992)).

For all experiments in Section 6, we obtained higher portfolio values when we allocated the amounts of capacity according to the methods developed by Herbots et al. (2008) for concurrent projects, than when we appointed the upper-bound capacity values to the projects. This indicates that even in multi-project environments where projects are often not scheduled purely in parallel or sequentially, the methods in Herbots et al. (2008) will still be of use and remain good guidelines to determine the extent to which projects should be downsized.

Allowing for concurrent planning transforms the one-machine job sequencing problem with random processing times to an m-parallel machine planning problem, with m the maximum number of jobs that is allowed to be processed in parallel. Setting such a bound m

on the number of concurrent projects has the advantage that it keeps project cycle times acceptable (Adler et al. 1995). Under these circumstances, the sequencing rules derived in Section 5 are no longer optimal, but could still form the basis of a heuristic that prioritizes the projects based on the calculated index.

8. Conclusions

This paper establishes the usefulness of the methods that determine the optimal capacity amounts for parallel projects for the case where projects are sequenced. Our simulation experiments indicate that even when projects are sequenced, we can increase the value of the project portfolio by restricting the allotted capacity. The capacity allocations determined for parallel projects can be used to set boundaries on the capacity we allocate to projects that are executed in sequence in case of unknown workloads. Moreover, the findings of this paper allow us to use capacity-allocation methods for parallel projects in multi-project environments, where a combination of projects planned in sequence and concurrently executed projects typically occurs.

When workloads are deterministic and the interest rate is zero, the optimal policy performs far better than the intuitive greedy algorithm for project portfolio selection. In case of reinvestment revenues, a sequencing rule based on the allocated amounts of capacity can be straightforwardly determined. The weighted discounted shortest expected processing time rule is optimal for deterministic workloads and a similar rule has been formulated for the case of stochastic workloads.

Appendix

Proof (Proposition 1): The proof of this proposition is based on a pairwise interchange argument. Assume that we are at time t and we execute schedule S in which job 1 immediately precedes job 2 ($1 \prec 2$). An adjacent pairwise interchange of job 1 and 2 results in schedule S'. The difference in expected reward between the two schedules results only from the contributions of job 1 and job 2. Table 6 shows the contributions of job 1 and 2 in schedule S and S' in case workloads are stochastic and drawn from a uniform distribution, with the expected duration $E[\hat{d}_k(\hat{M}_k)]$ of project $k \in \{1, 2\}$ obtained from Eq. (7).

From these results, we obtain that schedule S performs at least as well as schedule S'

Table 6: Contributions of job 1 and 2 in schedule S and S'.

$S : 1 \prec 2$	$S' : 2 \prec 1$
$\int_{\mu_1(1-\beta_1)}^{\hat{M}_1} e^{-i(p+t)} \frac{\alpha_1 C_1 \mu_1^{\gamma_1-1}}{2\beta_1}\, dp+$	$\int_{\mu_2(1-\beta_2)}^{\hat{M}_2} e^{-i(p+t)} \frac{\alpha_2 C_2 \mu_2^{\gamma_2-1}}{2\beta_2}\, dp+$
$\int_{\hat{M}_1}^{\mu_1(1+\beta_1)} e^{-i(\hat{M}_1+t)} \frac{\alpha_1 \mu_1^{\gamma_1-1}}{2\beta_1} (\frac{\hat{M}_1}{p})^{\gamma_1} C_1\, dp+$	$\int_{\hat{M}_2}^{\mu_2(1+\beta_2)} e^{-i(\hat{M}_2+t)} \frac{\alpha_2 \mu_2^{\gamma_2-1}}{2\beta_2} (\frac{\hat{M}_2}{p})^{\gamma_2} C_2\, dp+$
$\int_{\mu_2(1-\beta_2)}^{\hat{M}_2} e^{-i(E[\hat{d}_1(\hat{M}_1)]+p+t)} \frac{\alpha_2 C_2 \mu_2^{\gamma_2-1}}{2\beta_2}\, dp+$	$\int_{\mu_1(1-\beta_1)}^{\hat{M}_1} e^{-i(E[\hat{d}_2(\hat{M}_2)]+p+t)} \frac{\alpha_1 C_1 \mu_1^{\gamma_1-1}}{2\beta_1}\, dp+$
$\int_{\hat{M}_2}^{\mu_2(1+\beta_2)} e^{-i(\hat{M}_2+E[\hat{d}_1(\hat{M}_1)]+t)} \frac{\alpha_2 \mu_2^{\gamma_2-1}}{2\beta_2} (\frac{\hat{M}_2}{p})^{\gamma_2} C_2\, dp$	$\int_{\hat{M}_1}^{\mu_1(1+\beta_1)} e^{-i(\hat{M}_1+E[\hat{d}_2(\hat{M}_2)]+t)} \frac{\alpha_1 \mu_1^{\gamma_1-1}}{2\beta_1} (\frac{\hat{M}_1}{p})^{\gamma_1} C_1\, dp$

when

$$
\begin{aligned}
&e^{-it}(1-e^{-iE[\hat{d}_2(\hat{M}_2)]})\times\\
&\Big(\int_{\mu_1(1-\beta_1)}^{\hat{M}_1} \frac{e^{-i(p)}\alpha_1 C_1 \mu_1^{\gamma_1-1}}{2\beta_1}\,dp + \int_{\hat{M}_1}^{\mu_1(1+\beta_1)} \frac{e^{-i\hat{M}_1}\alpha_1\mu_1^{\gamma_1-1}}{2\beta_1}(\tfrac{\hat{M}_1}{p})^{\gamma_1} C_1\,dp\Big) \geq\\
&e^{-it}(1-e^{-iE[\hat{d}_1(\hat{M}_1)]})\times\\
&\Big(\int_{\mu_2(1-\beta_2)}^{\hat{M}_2} \frac{e^{-ip}\alpha_2 C_2 \mu_2^{\gamma_2-1}}{2\beta_2}\,dp + \int_{\hat{M}_2}^{\mu_2(1+\beta_2)} \frac{e^{-i\hat{M}_2}\alpha_2\mu_2^{\gamma_2-1}}{2\beta_2}(\tfrac{\hat{M}_2}{p})^{\gamma_2} C_2\,dp\Big).
\end{aligned}
$$

A best sequence for projects with stochastic workloads is thus obtained when we rank every project k in non-increasing order of the index in Eq. (6). □

References

Adler, P., Mandelbaum, A., Nguyen, V. & Schwerer, E. (1995). From project to process management: an empirically-based framework for analyzing product development time. *Management Science*, 41(3), pp 458–484.

Alidaee, B., Kochenberger, G. & Amini, M. (2001). Greedy solutions of selection and ordering problems. *European Journal of Operational Research*, 134, pp 203–215.

Blau, R. (1973). N-job, one machine sequencing problems under uncertainty. *Management Science*, 20(1), pp 101–109.

Brealey, R. & Myers, S. (2003). *Principles of corporate finance*. McGraw Hill, New York.

Chen, J. & Askin, R. (2009). Project selection, scheduling and resource allocation with time dependent returns. *European Journal of Operational Research*, 193, pp 23–34.

Chuzhoy, J., Ostrovsky, R. & Rabani, Y. (2001). Approximation algorithms for the job interval selection problem and related scheduling problems. *IEEE Symposium on Foundations of Computer Science*. pp 348–356. citeseer.ist.psu.edu/chuzhoy01approximation.html.

Cobb, C. & Douglas, P. (1928). A theory of production. *American Economic Review*, 18, pp 139–165.

Cooper, R., Edgett, S. & Kleinschmidt, E. (2001). Portfolio management for new product development: Results of an industry practices study. *R&D Management*, 31(4), pp 361–380.

Dantzig, G. (1957). Discrete variables extremum problems. *Operations Research*, 5, pp 266–277.

De, P., Ghosh, J. & Wells, C. (1991). On the minimization of the weighted number of tardy jobs with random processing times and deadline. *Computers & Operations Research*, 18(5), pp 457–463.

Engels, D., Karger, D., Kolliopoulos, S., Sengupta, S., Uma, R. & Wein, J. (2003). Techniques for scheduling with rejection. *Journal of Algorithms*, 49, pp 175–191.

Ghosh, J. (1997). Job selection in a heavily loaded shop. *Computers & Operations Research*, 24(2), pp 141.

Guerrero, H. & Kern, G. (1998). How to more effectively accept and refuse orders. *Production and Inventory Management Journal*, 4, pp 59–62.

Gupta, S., Kyparisis, J. & Ip, C. (1992). Project selection and sequencing to maximize net present value of the total return. *Management Science*, 38, pp 751–752.

Herbots, J., Herroelen, W. & Leus, R. (2008). Capacity allocation and downsizing decisions in project portfolio management. *Research Report KBI 0812*. Department of Decision Sciences and Information Management, Katholieke Universiteit Leuven, Belgium.

Kavadias, S. & Loch, C. (2004). *Project selection under uncertainty. Dynamically allocating resources to maximize value*. Kluwer.

Kerzner, H. (1997). *Project management: A systems approach to planning, scheduling, and controlling*. John Wiley & Sons.

Keskinocak, P. & Tayur, S. (2004). Due date management policies. *in* D. Simchi-Levi, S. Wu & Z. Shen (eds), *Handbook of Quantitative Supply Chain Analysis: Modeling in the E-Business Era*. Kluwer. Chapter: 12. pp 485–547.

Lewis, H. & Slotnick, S. (2002). Multi-period job selection: Planning work loads to maximize profit. *Computers & Operations Research*, 29, pp 1081–1098.

Lu, L., Zhang, L. & Yuan, J. (2008). The unbounded parallel batch machine scheduling with release dates and rejection to minimize makespan. *Theoretical Computer Science*, 396, pp 283–289.

Pinedo, M. (2008). *Scheduling - Theory, algorithms, and systems*. Springer Science, NY.

Rom, W. & Slotnick, S. (2008). Order acceptance using genetic algorithms. *Computers & Operations Research*, . To appear.

Roundy, R., Chen, D., Chen, P., Cakanyildirim, M., Freimer, M. & Melkonian, V. (2005). Capacity-driven acceptance of customer orders for a multi-stage batch manufacturing system: models and algorithms. *IIE Transactions*, 37, pp 1093–1105.

Sharpe, P. & Keelin, T. (1998). How SmithKline Beecham makes better resource-allocation decisions. *Harvard Business Review*, pp 45–57.

Slotnick, S. & Morton, T. (1996). Selecting jobs for a heavily loaded shop with lateness penalties. *Computers & Operations Research*, 23, pp 131–140.

Slotnick, S. & Morton, T. (2007). Order acceptance with weighted tardiness. *Computers & Operations Research*, 34(10), pp 3029–3042.

Tushman, M., Anderson, P. & O'Reilly, C. (1997). *Managing strategic innovation and change*. Oxford University Press. Chapter: Technology cycles, innovation streams, and ambidextrous organizations: Organization renewal through innovation streams and strategic change. pp 3–23.

Wheelwright, S. & Clark, K. (1992). Creating project plans to focus product development. *Harvard Business Review*, pp 70–82.

Yang, B. & Geunes, J. (2007). A single resource scheduling problem with job-selection flexibility, tardiness costs and controllable processing times. *Computers & Industrial Engineering*, 53, pp 420–432.